

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Implementasi

Dalam tahap ini dijelaskan mengenai implementasi perangkat lunak. Perangkat lunak yang dibangun dikembangkan dengan menggunakan bahasa pemrograman Microsoft Visual Basic 6.0 dan menggunakan ASP untuk pemrograman *Online*, dan untuk *Database* menggunakan Microsoft Access 2000

4.1.1 Kebutuhan sistem

Aplikasi pada sistem pakar untuk mendiagnosa penyakit flu burung ini memerlukan perangkat lunak (*software*) dan perangkat keras (*hardware*), agar dapat berjalan sesuai dengan yang diharapkan.

Adapun perangkat lunak yang digunakan yaitu:

- Sistem operasi Windows 95/Me/2000/XP
- Aplikasi program adalah Microsoft Visual Basic 6.0
- Database* untuk menyimpan data adalah Microsoft Access 2000
- Internet Information Service (IIS)
- Browser yang digunakan adalah Internet Explorer 5.0 atau yang lebih baru.
- Script yang digunakan untuk mengakses *Database* adalah *script* ASP.

Perangkat keras yang digunakan yaitu:

- Processor Pentim II atau lebih.
- Memory 32 Mb
- Harddisk 10 Gb
- VGA 8 Mb.

- e. Monitor
- f. Keyboard dan mouse

4.1.2 Instalasi Program dan Pengaturan Sistem

Pengembangan aplikasi sistem pakar untuk mendiagnosa penyakit flu burung membutuhkan perangkat lunak yang sudah terinstal, adapun tahapan instalasi dan pengaturan (*setting*) sistem yaitu:

1. Install sistem operasi Windows 98/Me/2000/XP
2. Install aplikasi program Microsoft Visual Basic 6.0
3. Install aplikasi *Database* Microsoft Access 2000
4. Install Internet Information Service (IIS)
5. Install Seagate Crystal Report 8.0

4.1.3 Penjelasan Pemakaian Program

Sistem pakar untuk mendiagnosa penyakit flu burung ini dapat dijalankan setelah dilakukan tahap-tahap instalasi program seperti diatas. *User Interface* yang ditampilkan antara lain.

1. Login

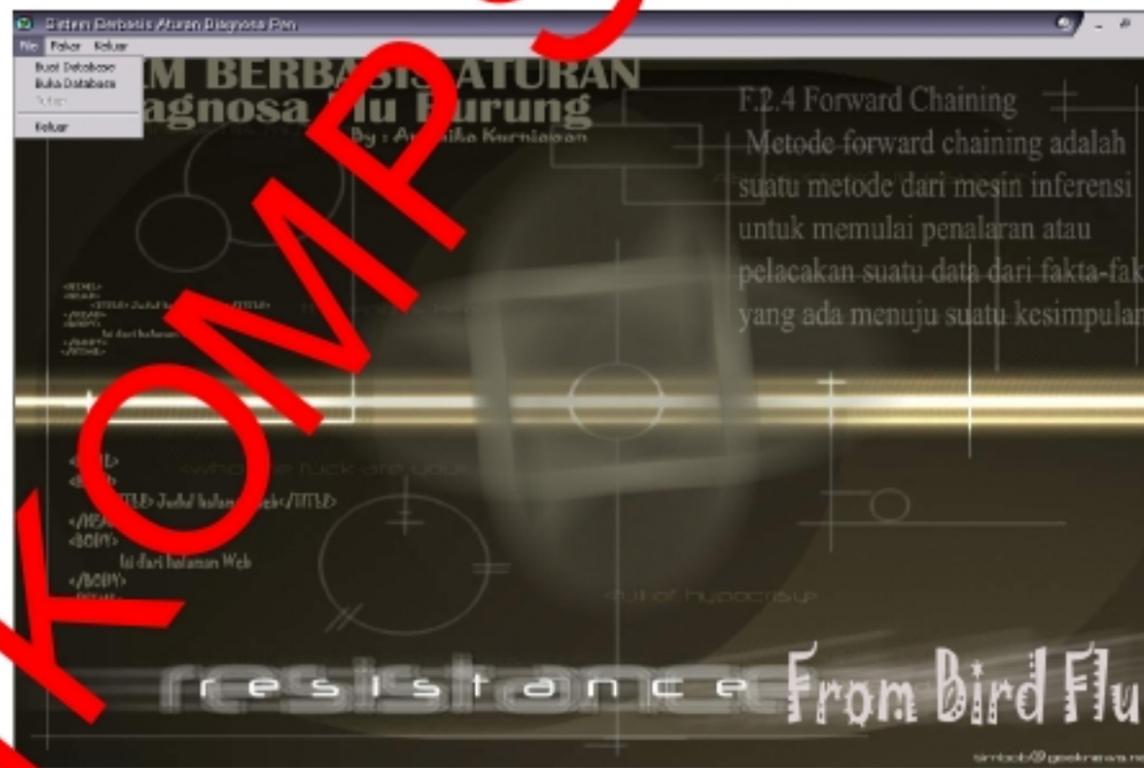
Form ini berfungsi untuk login agar bisa masuk ke dalam sistem. Form login ini dapat dilihat pada gambar 4.1. Menu login ini digunakan untuk masuk ke dalam sistem. *UserID* dan *Password* yang digunakan akan menentukan siapa yang berhak untuk mengoperasikan sistem, yaitu pakar dan *User*.



Gambar 4.1 Menu Utama

2. Tampilan Utama

Tampilan utama pada sistem ini adalah tampilan form awal yang dapat dilihat pada Gambar 4.2.



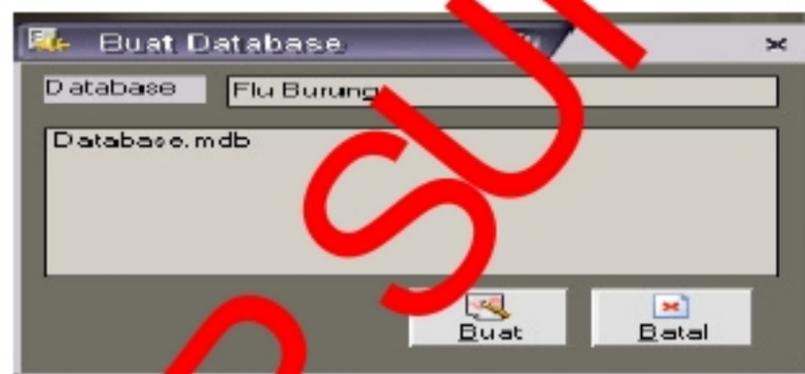
Gambar 4.2. Tampilan Utama

Tampilan utama ini terdiri dari menu-menu dan sub menu antara lain: menu *File* yang terdiri dari sub menu *buat Database*, *buka Database*, *tutup*

Database dan keluar. Menu pakar terdiri dari sub menu yaitu: desain *Treeview*, rancang aturan dan set web *Database*. Menu *Password* terdiri dari sub menu ubah *UserId/Password*, maintain *User* untuk menambahkan *User* baru dan sub menu *LogOff* untuk keluar dari program.

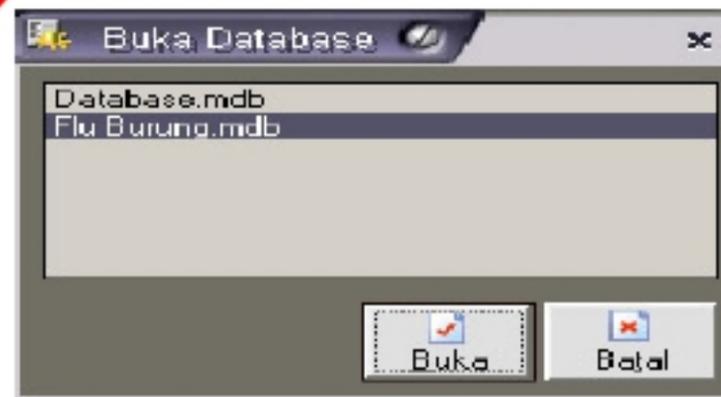
3. Tampilan Pakar untuk Aturan *Treeview*

Pada langkah awal sebelum membuat desain *Treeview* adalah dengan membuat *Database* dulu terlebih dahulu, yang digunakan untuk menyimpan *Database* yang akan digunakan oleh pakar dalam merancang *Treeview*, seperti gambar 4.3



Gambar 4.3 Desain Membuat *Database* baru

Pada Gambar 4.3 setelah *Database* baru tersimpan, proses selanjutnya adalah open *Database*, seperti pada gambar 4.4.



Gambar 4.4 Desain *Open Database*

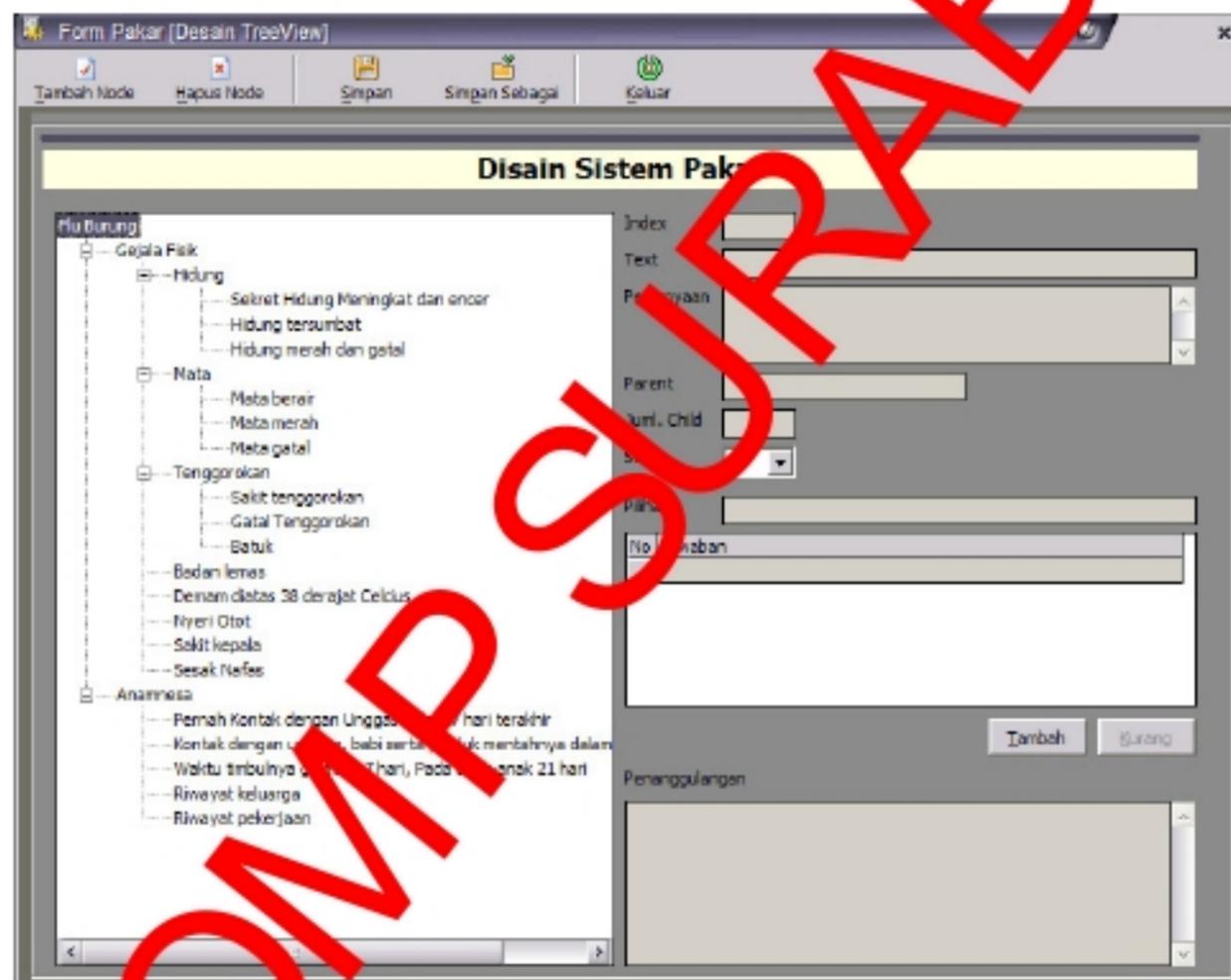
Proses *open Database* pada Gambar 4.4 menentukan nama *Database* yang akan digunakan untuk membuat *Treeview*.

Berikut form untuk membuat *Treeview*. Desain *Treeview* yang akan digunakan untuk proses diagnosa penyakit flu burung, dimulai dengan proses tambah *Node* seperti pada Gambar 4.5. *Node* awal pada *Treeview* merupakan set pertama pada *Dependency Diagram* yang sudah kita buat di awal.

Gambar 4.5 Desain Tree View baru

Pada Gambar 4.5 untuk desain *Treeview* langkah awal yang harus dilakukan yaitu dengan membuat *Node* terlebih dahulu sesuai dengan kebutuhan pakar dengan menekan tombol tambah *Node*.

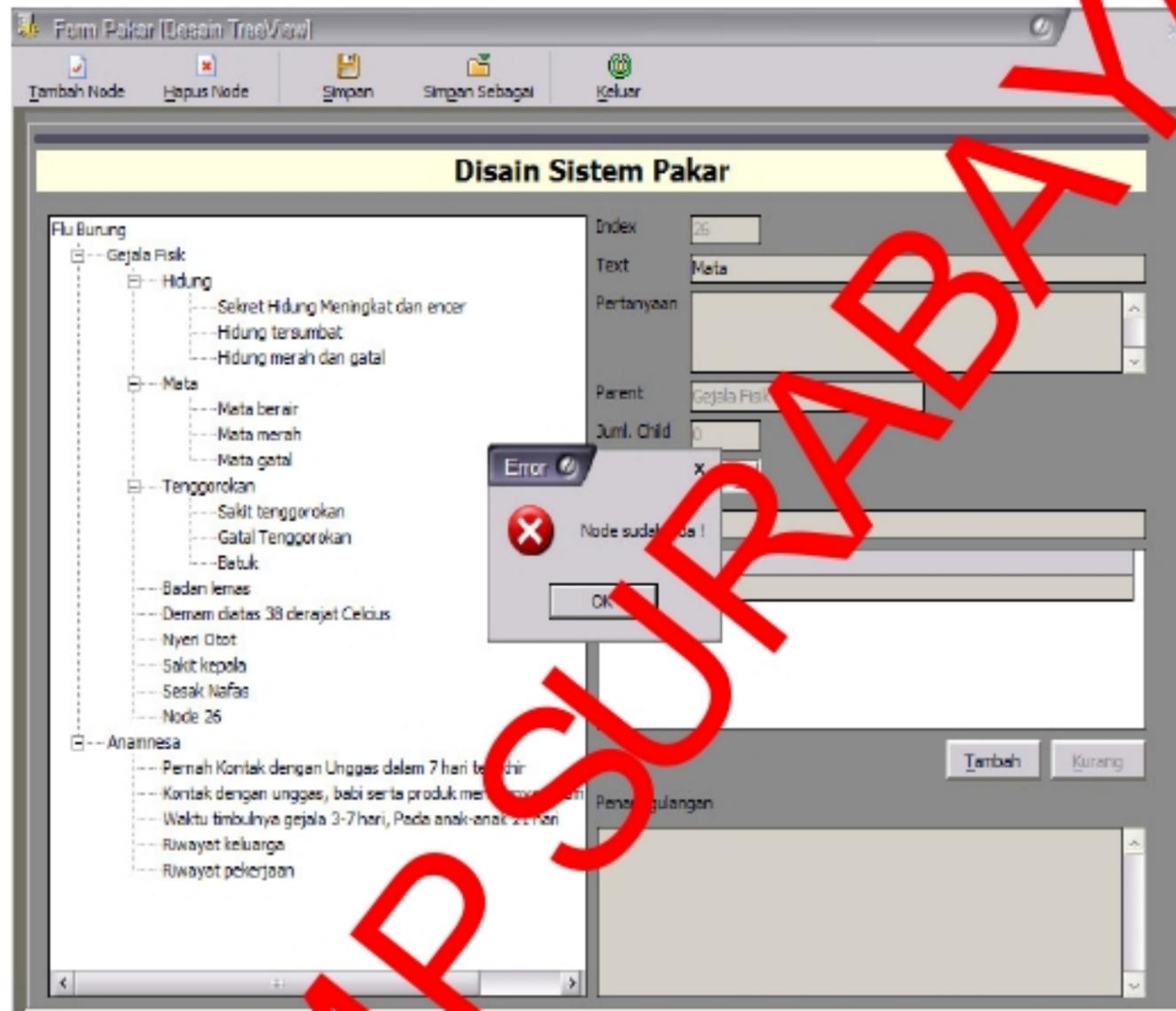
Langkah selanjutnya adalah penentuan parameter *Treeview* yang disesuaikan dengan *Dependency Diagram*. Ini digunakan untuk menentukan hubungan antara faktor-faktor penting yang mempengaruhi dalam pemberian suatu rekomendasi penyakit. *Treeview* juga berisi aturan-aturan dan jawaban yang digunakan untuk memudahkan pada saat proses Verifikasi yang telah kita buat sebelumnya, seperti pada Gambar 4.6.



Gambar 4.6 Penentuan Parameter Pada Desain *Treeview* Baru

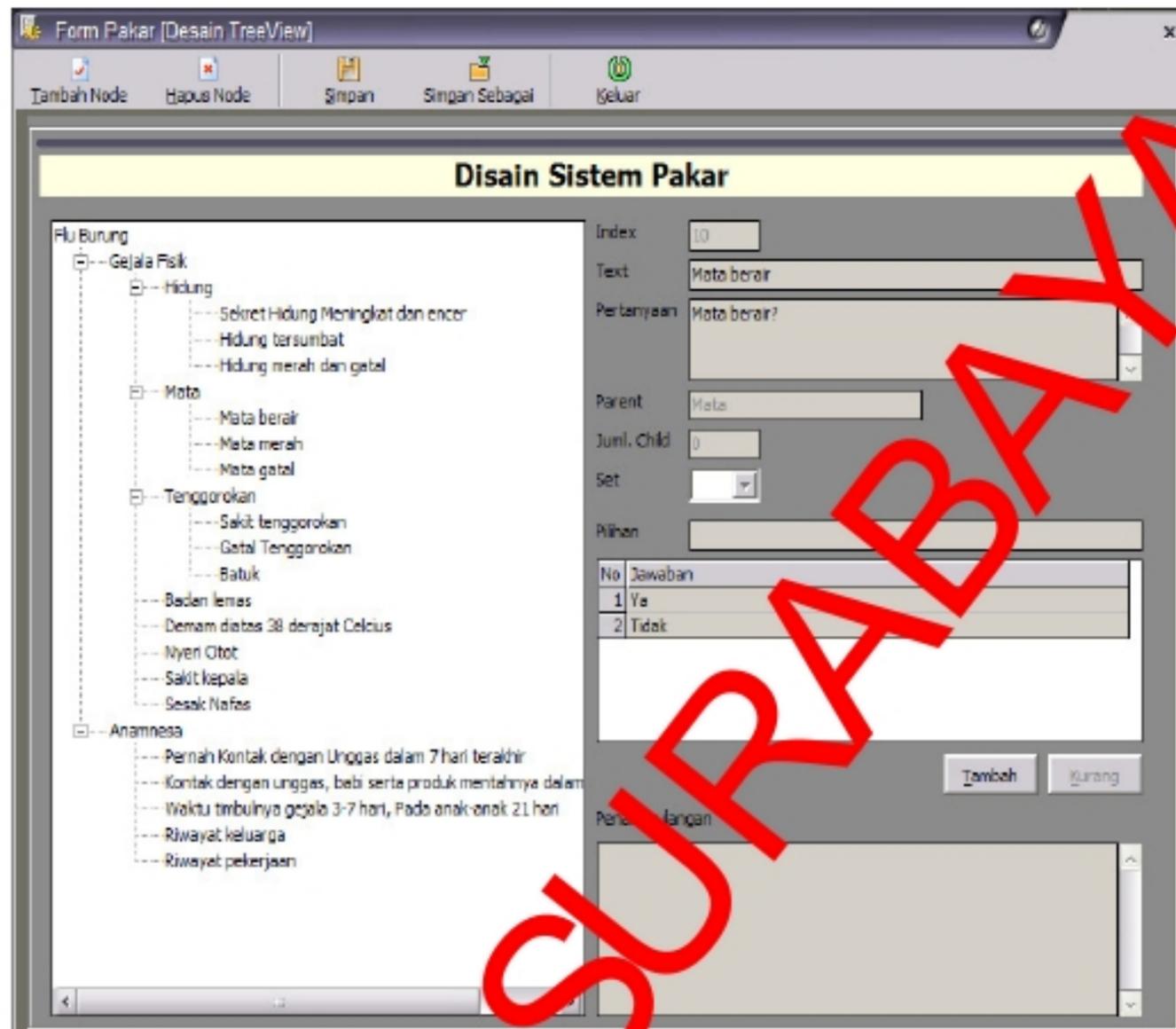
Penentuan parameter pada Gambar 4.6 dapat diubah jika pakar mendidaki perubahan, untuk mengganti text dari format “*Node*” menjadi sebuah parameter, isikan pada area *Text*, dan parameter ini dapat juga dihapus, sedangkan untuk *Child* pada *Node*, merupakan sub parameter yang secara otomatis ada apabila kita menambahkan *Node* pada tiap-tiap set.

Pada proses input parameter diberikan fasilitas untuk validasi parameter dimana nama *Node* tidak boleh sama, seperti pada Gambar 4.7.



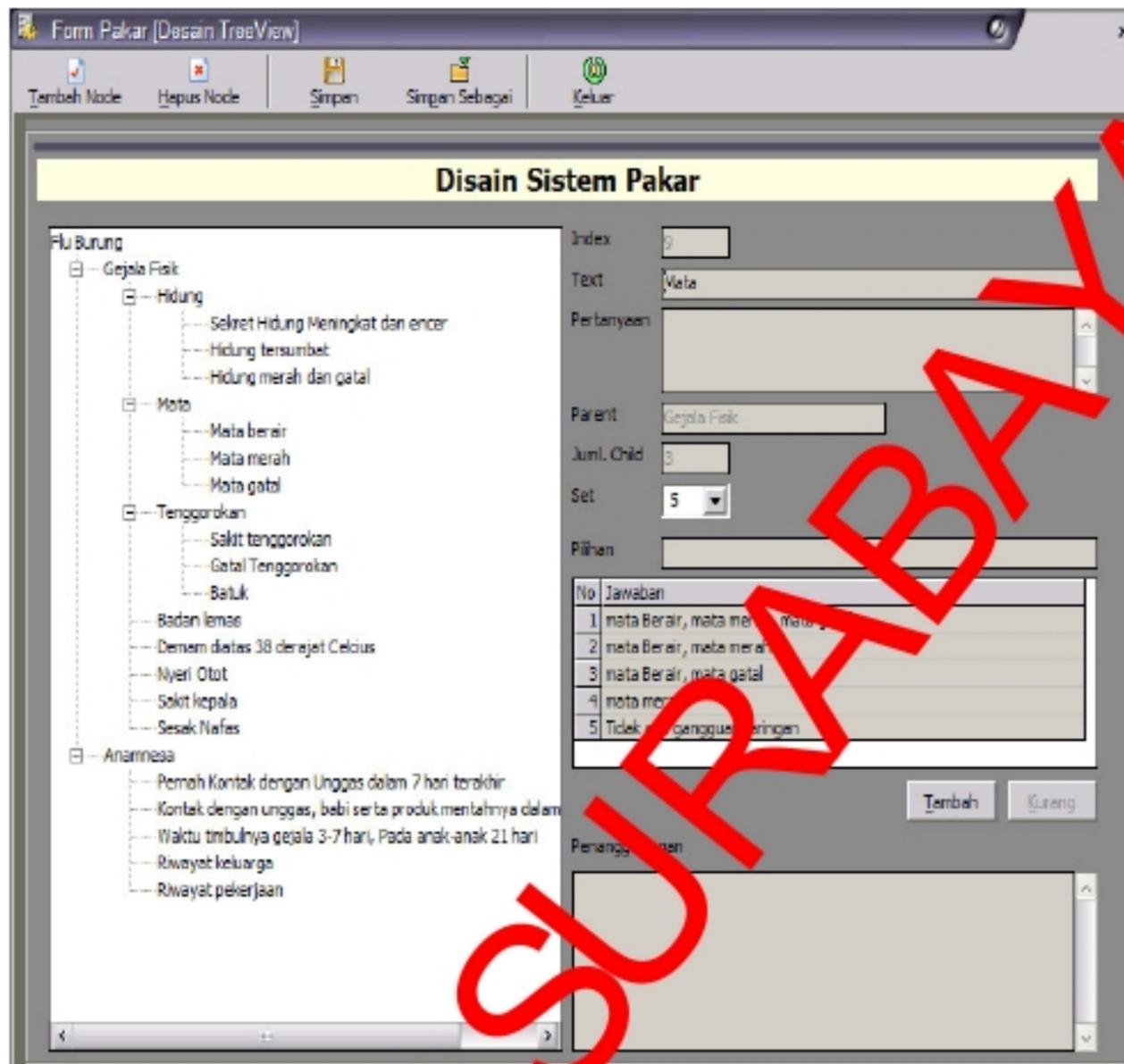
Gambar 4.7 Validasi *Input* Parameter

Form berikut menunjukkan langkah selanjutnya yaitu menentukan pertanyaan dan *Option* seperti pada gambar 4.8. Pada Gambar 4.8 dijelaskan bahwa sistem memberikan pertanyaan pada *Node-Node* yang tidak memiliki *Children*, ketikkan pertanyaan pada area *Text* pertanyaan. Untuk memberikan pilihan jawaban pada tiap-tiap *Node*, ketikkan pilihan jawaban pada area *Text* pilihan lalu tekan tombol tambah memasukkan kedalam list pilihan yang nantinya akan muncul secara otomatis pada form rancang aturan.



Gambar 4.8 Menentukan Pertanyaan dan *Option*

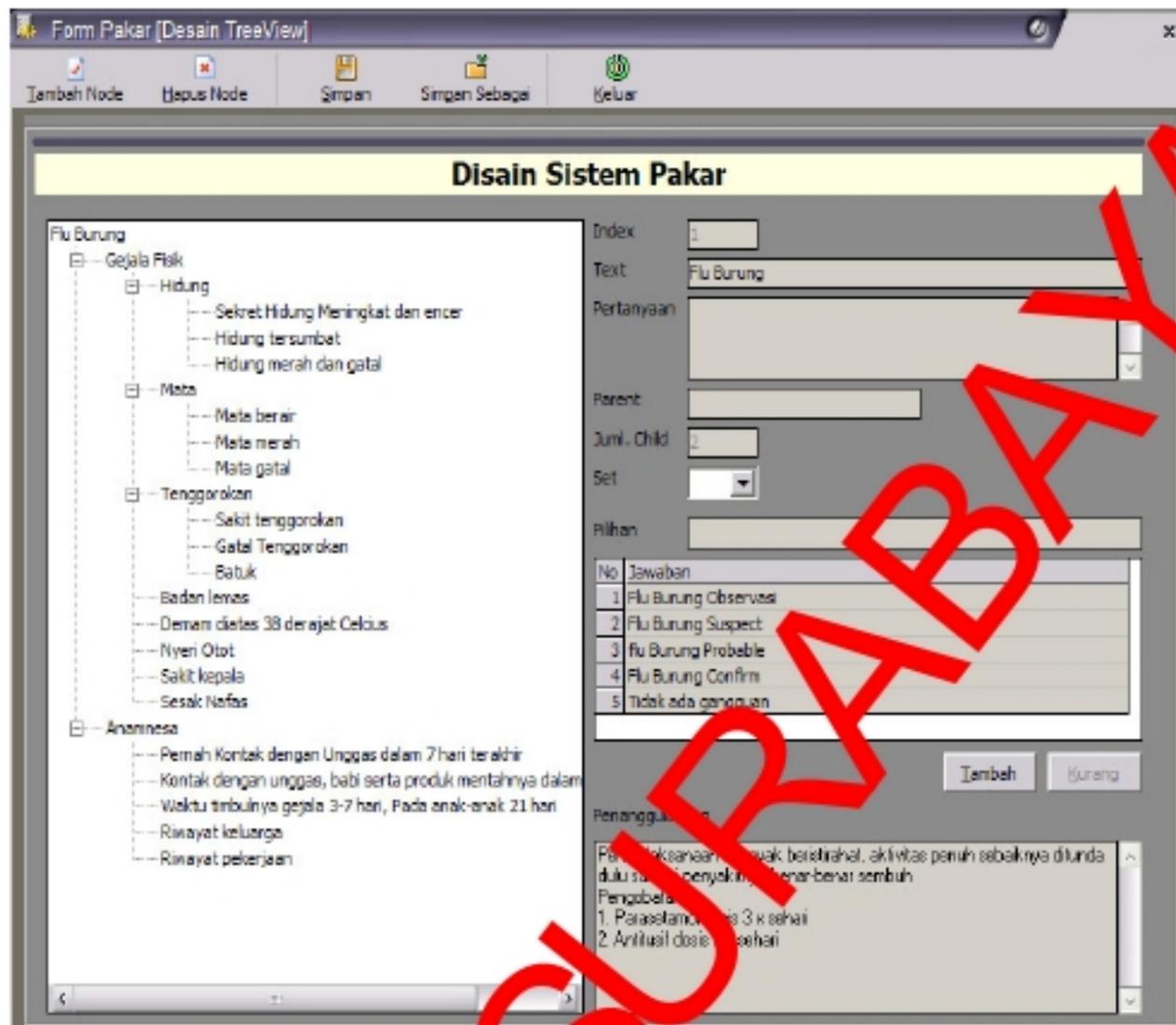
Poses selanjutnya adalah menentukan *Option* jawaban untuk tiap set berdasarkan pada *Dependency Diagram* yang telah kita buat seperti pada Gambar 4.9. *Option* jawaban dari tiap-tiap set dan pertanyaan ini juga harus disesuaikan dengan redaksi tabel yang sudah kita buat sebelumnya. Dan *Option-Option* akan muncul otomatis pada saat kita merancang aturan dan akan melakukan verifikasi. Tujuan dari *Option* ini adalah sebagai *conclution* dari *Rule-Rule* yang akan terbentuk secara otomatis setelah kita melakukan Verifikasi dari tiap-tiap set yang ada.



Gambar 4.9 Menentukan *Option* Jawaban Tiap Set

Penentuan *option* jawaban pada gambar 4.9 dapat diubah jika pakar menghendaki perubahan, selain itu *Option* juga dapat ditambah dan dihapus.

Proses selanjutnya adalah penentuan *Text* pengobatan sesuai dengan nama jenis penyakit flu burung, seperti pada Gambar 4.10.

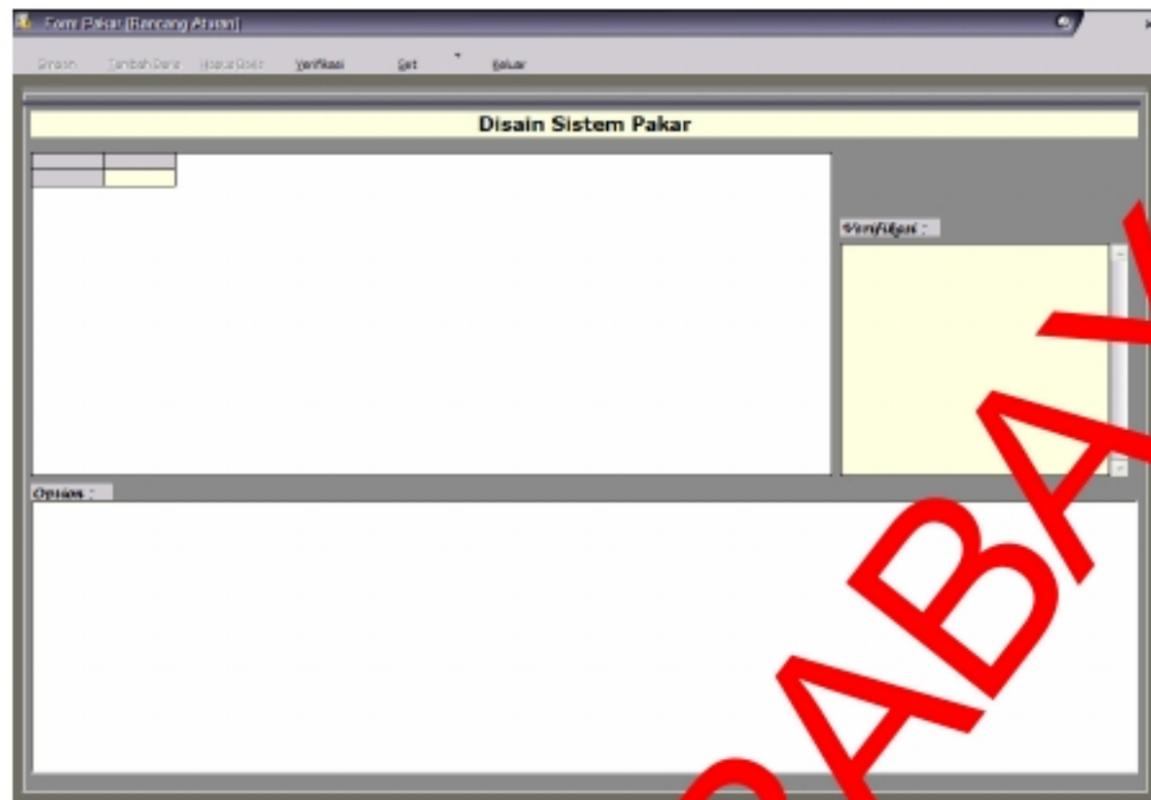


Gambar 4.10 Menentukan Pengobatan Penyakit Flu Burung

Pada Gambar 4.10 untuk memberikan pengobatan tiap-tiap penyakit flu burung yang terdapat dalam pilihan jawaban pada *Node root*, pilih salah satu penyakit flu burung dari *List* pilihan kemudian ketikkan pengobatan dari penyakit flu burung yang dipilih tersebut. Untuk mengisi pengobatan pada penyakit flu burung lain lakukan dengan cara yang sama seperti diatas.

4. Tambilan Pakar untuk List Aturan

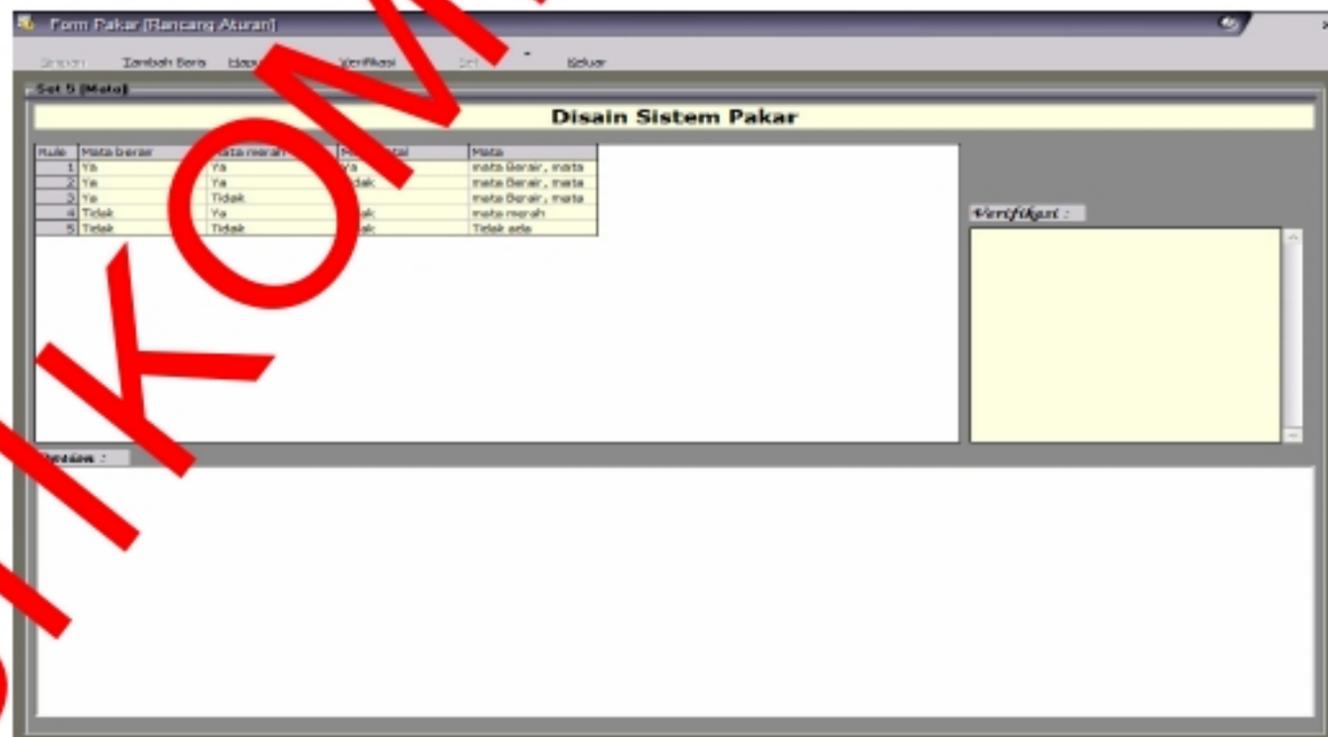
Pada tampilan berikut adalah membuat *List Aturan* dengan cara memasukkan parameter yang sudah ada berdasarkan pada hasil reduksi yang dilakukan secara manual, yang dimulai dengan proses menentukan set seperti pada Gambar 4.11.



Gambar 4.11 Menentukan Set Aturan

Pada Gambar 4.11 untuk membuat list aturan langkah awal yang harus dilakukan yaitu dengan menentukan set terlebih dahulu.

Proses selanjutnya adalah memasukkan aturan dengan mengklik pilihan yang ada di *Option*, seperti pada gambar 4.12.

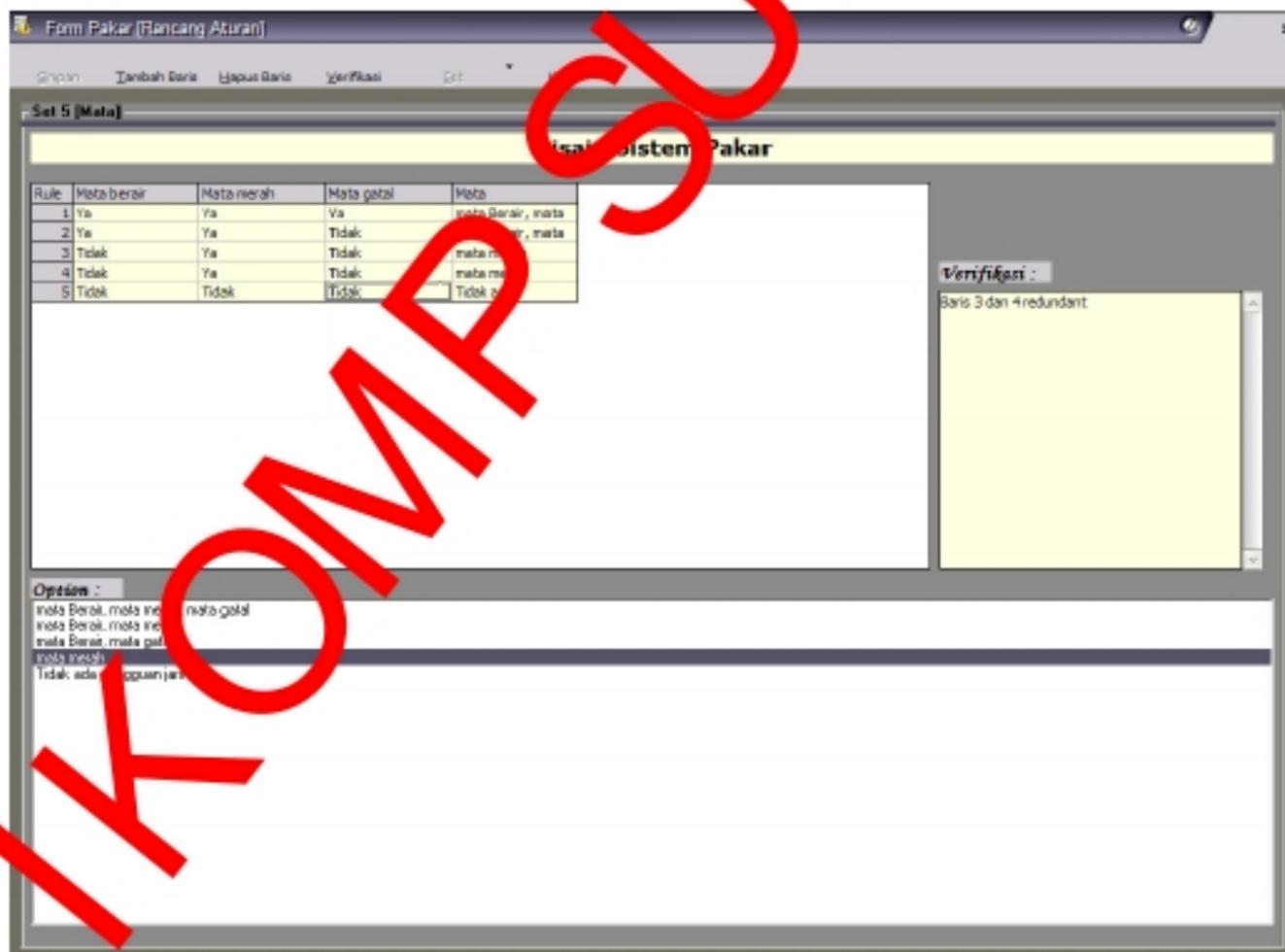


Gambar 4.12 Menentukan List Aturan Berdasarkan *Option*

Inputkan kombinasi dari parameter-parameter agar menjadi sebuah aturan. Klik salah satu *Cell* maka akan muncul pilihan jawaban dalam list *Option*. Untuk menambah baris klik tombol tambah baris sedangkan untuk menghapus baris klik baris yang akan dihapus terlebih dahulu kemudian klik tombol hapus baris. Penentuan *Option* untuk list aturan pada Gambar 4.12 belum dapat disimpan apabila pada proses Verifikasi masih terdapat *Rule* yang *Error*.

Poses selanjutnya adalah proses Verifikasi untuk *Premis* dan *Conclusion* seperti contoh berikut:

1. *Redundant Rule* yaitu jika dua *Rule* atau lebih mempunyai *Premise* dan *Conclusion* yang sama, seperti terlihat pada gambar 4.13.



Gambar 4.13 Hasil Verifikasi *Redundant Rule*

Pada Gambar 4.13 hasil Verifikasi menunjukkan pada baris 1 dan 4 adalah *redundant Rule*, karena mempunyai *Premis* dan *Conclusion* yang sama.

3. *Conflicting Rule* yaitu dua buah *Rule* atau lebih mempunyai *Premis* yang sama, tetapi mempunyai *Conclusion* yang berlawanan seperti pada Gambar 4.14

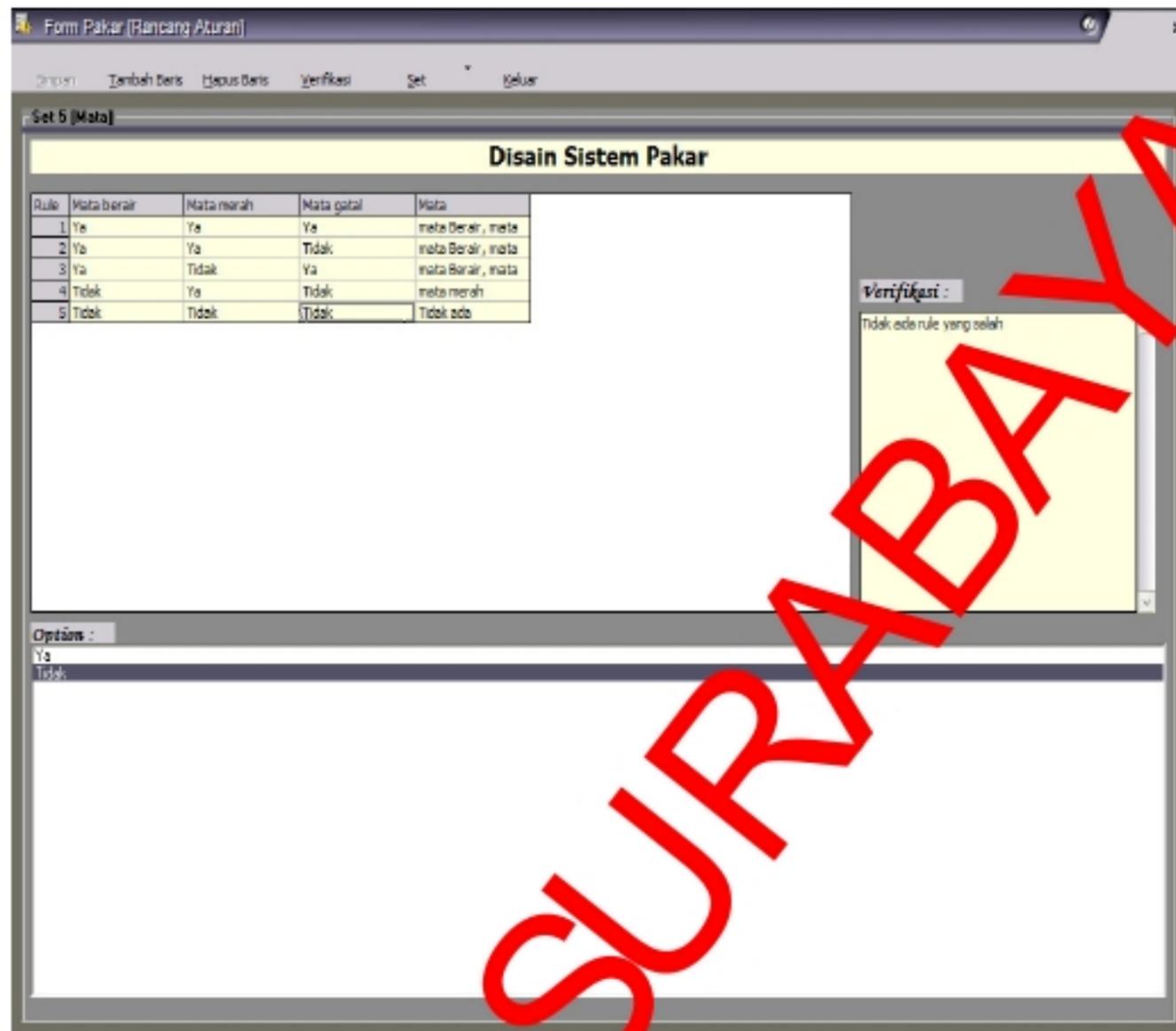
The screenshot shows a software window titled "Form Pakar (Rancang Aturan)" with a menu bar containing "Ganti", "Tambah Baris", "Hapus Baris", "Verifikasi", "Print", and "Keluar". Below the menu is a section titled "Set 5 (Mata)" and "Disain Sistem Pakar". It contains a table with 5 columns: "Rule", "Mata benar", "Mata merah", "Mata putih", and "Mata". The table has 5 rows of data. Below the table is an "Options" section with radio buttons for "Ya" and "Tidak". A yellow message box on the right side of the window displays the text "Baris 2 dan 4 conflicting".

Rule	Mata benar	Mata merah	Mata putih	Mata
1	Ya	Ya	Ya	mata Berak, mata
2	Ya	Ya	Tidak	mata Berak, mata
3	Tidak	Ya	Tidak	mata merah
4	Ya	Ya	Tidak	Tidak ada
5	Tidak	Tidak	Tidak	Tidak ada

Gambar 4.14 Hasil Verifikasi *Conflicting Rule*

Pada Gambar 4.14 hasil Verifikasi menunjukkan pada baris 2 dan 4 adalah *conflicting Rule*, karena dua *Rule* atau lebih mempunyai *Premis* yang sama tetapi mempunyai *conclusion* yang berlawanan.

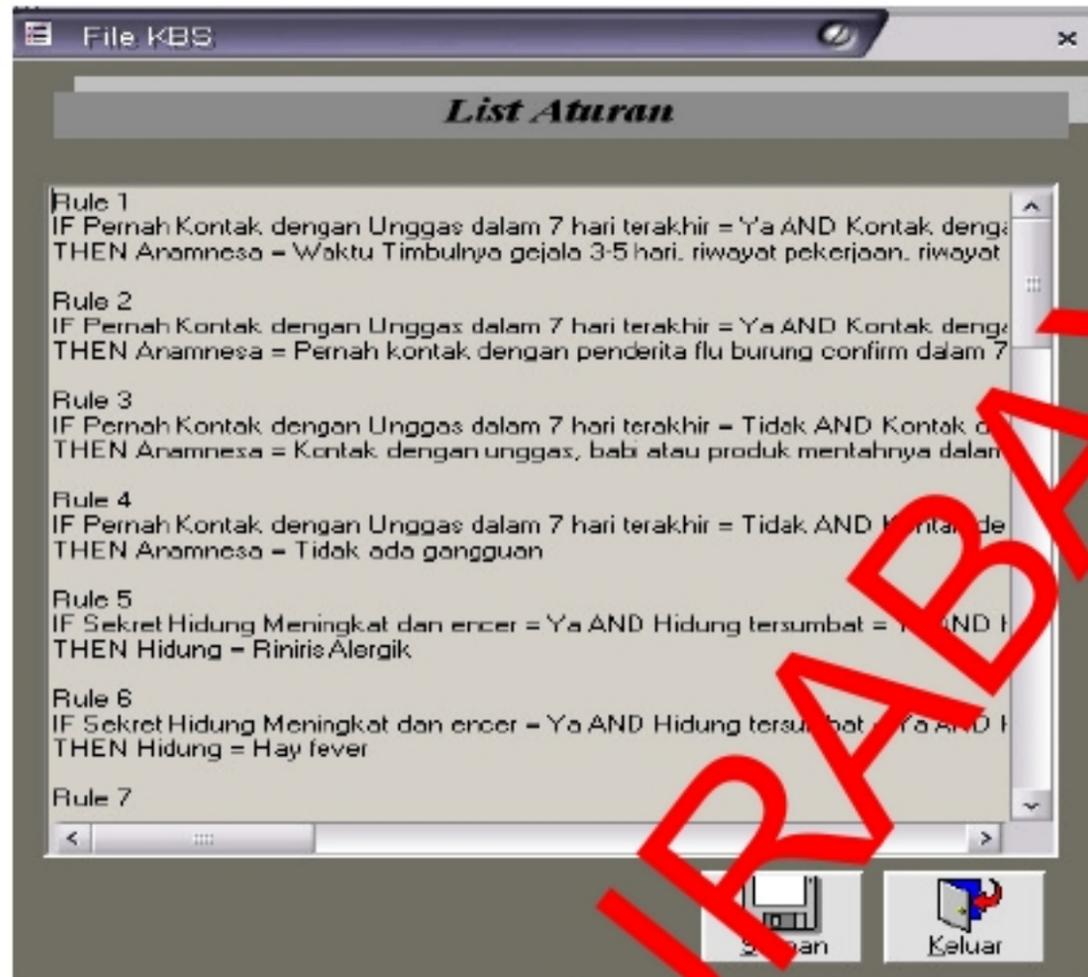
Sedangkan jika hasil Verifikasi menunjukkan tidak terdapat *Rule* yang *error*, seperti pada Gambar 4.15.



Gambar 4.15 Hasil Verifikasi *Rule* Tidak Error

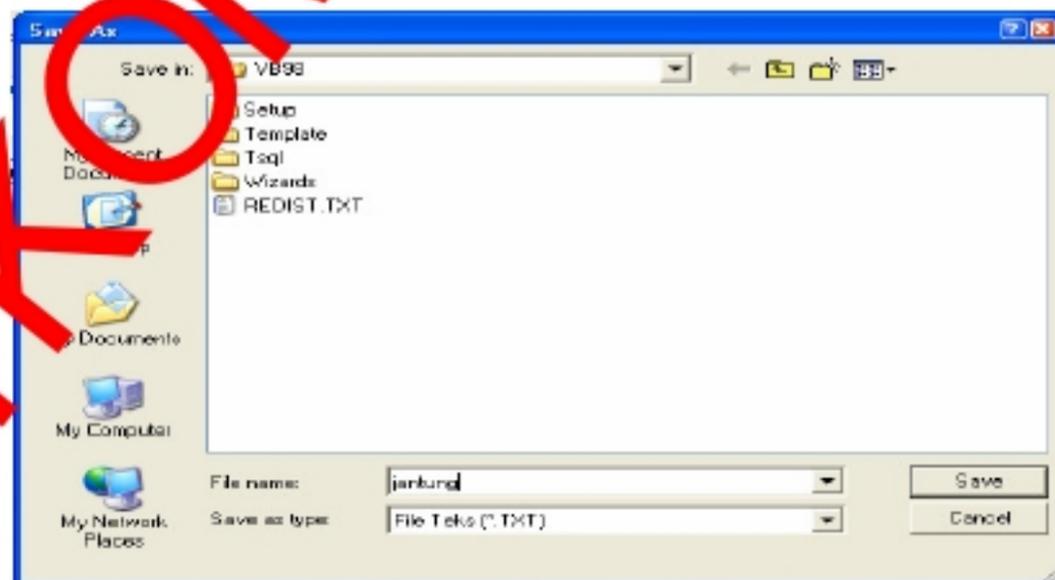
Pada Gambar 4.15 jika semua *Rule* telah benar maka akan muncul pesan “tidak ada *Rule* yang salah” maka tombol set akan aktif dan pakar dapat memilih set yang lain untuk dirancang aturan-aturannya. Tombol simpan tidak akan aktif jika semua set belum terverifikasi. Bila semua set telah terverifikasi maka tombol simpan akan aktif dan data-data dapat disimpan dengan mengklik tombol simpan.

Setelah data-data tersimpan maka aturan yang telah dibuat berdasarkan hasil verifikasi dapat dilihat pada *File* KBS, seperti pada Gambar 4.16.



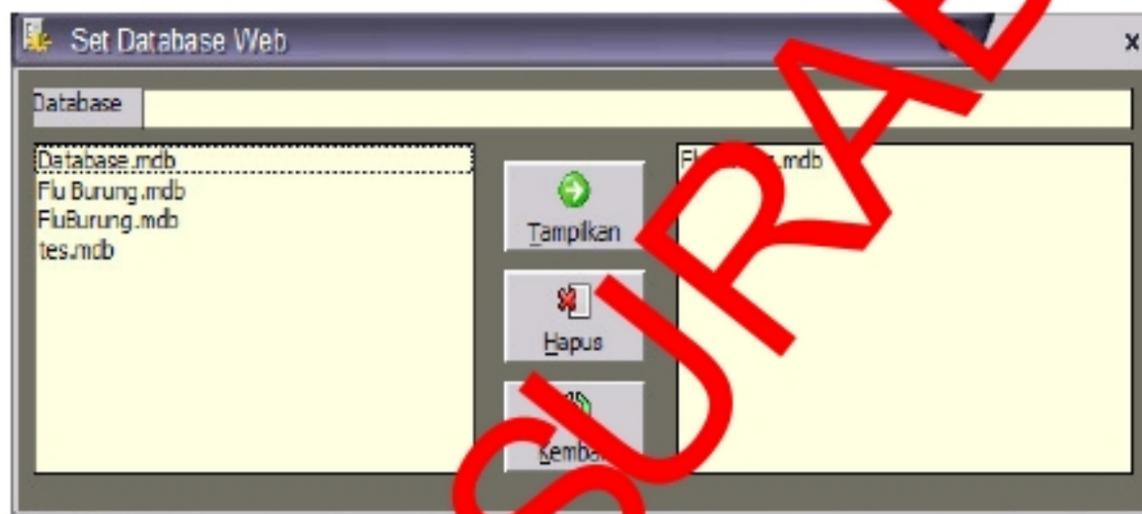
Gambar 4.16 Hasil *File Text*

Pada Gambar 4.16 menampilkan aturan-aturan yang terbentuk dari rancangan aturan tadi. Klik tombol simpan pada form *List Aturan* untuk menyimpan list aturan tadi dalam format *File .Txt*, seperti pada Gambar 4.17.



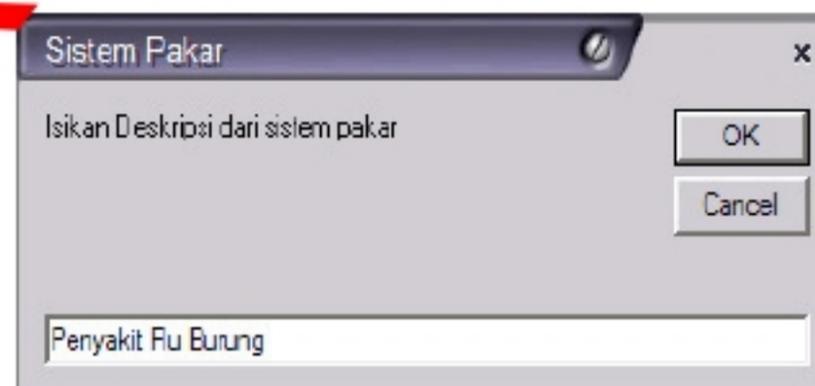
Gambar 4.17 Simpan *File Text*

Apabila semua proses untuk mendesain aturan sudah dilakukan maka langkah selanjutnya adalah menentukan set *Database* yang nantinya digunakan oleh *User* untuk proses konsultasi diagnosa penyakit flu burung, seperti pada Gambar 4.18. Pada proses set *Database* ini kita bisa memasukkan lebih dari satu *Database* penyakit flu burung sehingga nantinya *User* dapat memilih *Database* yang akan digunakan untuk proses konsultasi.



Gambar 4.18 Set *Database Web* untuk Proses Konsultasi

Setelah kita menentukan *Database* yang akan dipilih proses selanjutnya adalah memberikan deskripsi pada *Database* yang akan kita set. Hal ini digunakan untuk memberikan fasilitas kepada *User* untuk memilih jenis konsultasi sesuai *Database* yang telah dibuat oleh pakar seperti pada gambar 4.19.



Gambar 4.19 Deskripsi Pemilihan Konsultasi *Web Database*

4. Tampilan Untuk *User* Secara *Online*

Pada tampilan ini adalah proses konsultasi yang dilakukan *User* secara online. Pada halaman utama sistem pakar untuk diagnosa penyakit flu burung menggambarkan tentang *Login* yang digunakan oleh *User*, seperti Gambar 4.20. Tombol *Submit* digunakan untuk *User* yang telah terdaftar dan langsung bisa memasuki sistem. Tombol *Reset* digunakan untuk membatalkan *login*.



Gambar 4.20 Menu *Login* Proses Konsultasi

Sedangkan untuk *User* yang belum terdaftar maka *User* baru bisa mendaftar dengan melakukan klik pada petunjuk untuk non anggota daftar disini, seperti pada Gambar 4.21. Setelah dilakukan klik maka akan muncul *form* registrasi untuk *User* baru, dan *User* bisa mengisi *Form* yang ada. Tombol *Submit* untuk menyetujui dan tombol *Reset* untuk membatalkan

The screenshot shows a web browser window with the address bar displaying 'http://localhost/test/register.asp'. The page header features a logo with 'COMPANY NAME' and the title 'WASPADA PENYAKIT MILENIUM'. The main content area is titled 'New User Registration Form' and contains the following fields: 'Nama Login', 'Nama', 'Alamat', 'Password', 'Tgl Lahir' (with a dropdown for DD/MM/YYYY), and 'Jenis Kelamin' (with a dropdown for Laki-Laki). Below the fields are 'Submit' and 'Reset' buttons, and a 'Kembali' link at the bottom.

Gambar 4.21 Form Registrasi User Baru

Setelah *Login* berhasil proses selanjutnya adalah *User* dapat memilih jenis konsultasi yang disediakan oleh sistem seperti pada Gambar 4.22.

The screenshot shows a web browser window with the address bar displaying 'http://localhost/test/default.asp'. The page header is identical to Gambar 4.21. The main content area is titled 'Pilih Jenis Konsultasi' and contains the text 'Selamat Datang,', a greeting, and a dropdown menu for 'Pilih Jenis Konsultasi' with options like 'Pilih Jenis Konsultasi', 'Pilih Jenis Konsultasi', and 'Pilih Jenis Konsultasi'. There is a 'Tampilkan History Konsultasi Anda' link and a 'Logout' button at the bottom.

Gambar 4.22 Menu Pilih Jenis Konsultasi

Proses selanjutnya adalah *Input* jawaban. Pada proses ini *User* menjawab semua pertanyaan yang di ajukan oleh sistem, seperti pada Gambar 4.23.

Gambar 4.23 *Input Jawaban Pertanyaan*

Pada Gambar 4.23, *User* menjawab semua pertanyaan yang diberikan oleh sistem sampai selesai. Setelah proses *input* jawaban selesai maka *User* dapat melakukan klik pada tombol *Submit* jika yakin pada jawaban telah benar dan tombol *Reset* jika ingin membatalkannya.

Untuk mengetahui hasil diagnosa penyakit flu burung beserta pengobatannya dapat dilihat pada Gambar 4.24.

Gambar 4.24 Hasil Diagnosa Penyakit Flu Burung

Setelah tombol *Submit* di klik maka akan muncul jenis penyakit flu burung yang diderita pasien, beserta pengobatan yang perlu dilakukan. Pakar dapat memberikan rekomendasi obat yang perlu di konsumsi oleh pasien.

5. Tampilan Laporan *Medical record*

Tampilan untuk laporan hasil konsultasi *User* yang menderita penyakit flu burung digunakan sebagai *Medical Record* atau rekam medis *User* yang telah melakukan konsultasi seperti pada gambar 4.25. Pada proses ini hanya dapat dilakukan oleh administrator dalam hal ini pakar.

No.Registrasi	Nama	Alamat	Umur	J.Kelamin	Tgl.Periksa	Penyakit
001	xxx	xxx	5	Laki-laki	25-Jul-2004	Tidak Diderita
002	xxx	xxx	24	Laki-laki	25-Jul-2004	DBD I
002	xxx	xxx	24	Laki-laki	25-Jul-2004	Tidak Diderita
001	xxx	xxx	5	Laki-laki	25-Jul-2004	Tidak Diderita
001	xxx	xxx	5	Laki-laki	25-Jul-2004	DBD I
001	xxx	xxx	5	Laki-laki	25-Jul-2004	Tidak Diderita
001	xxx	xxx	5	Laki-laki	25-Jul-2004	Tidak Diderita
001	xxx	xxx	5	Laki-laki	25-Jul-2004	Chorussia Chronitum

Gambar 4.25. Laporan Rekam Medis

2 Evaluasi

Untuk pengujian logika dan evaluasi hasil akhir proses Verifikasi dan proses inferensi ini maka digunakan beberapa data yang bersifat pasti, dimana

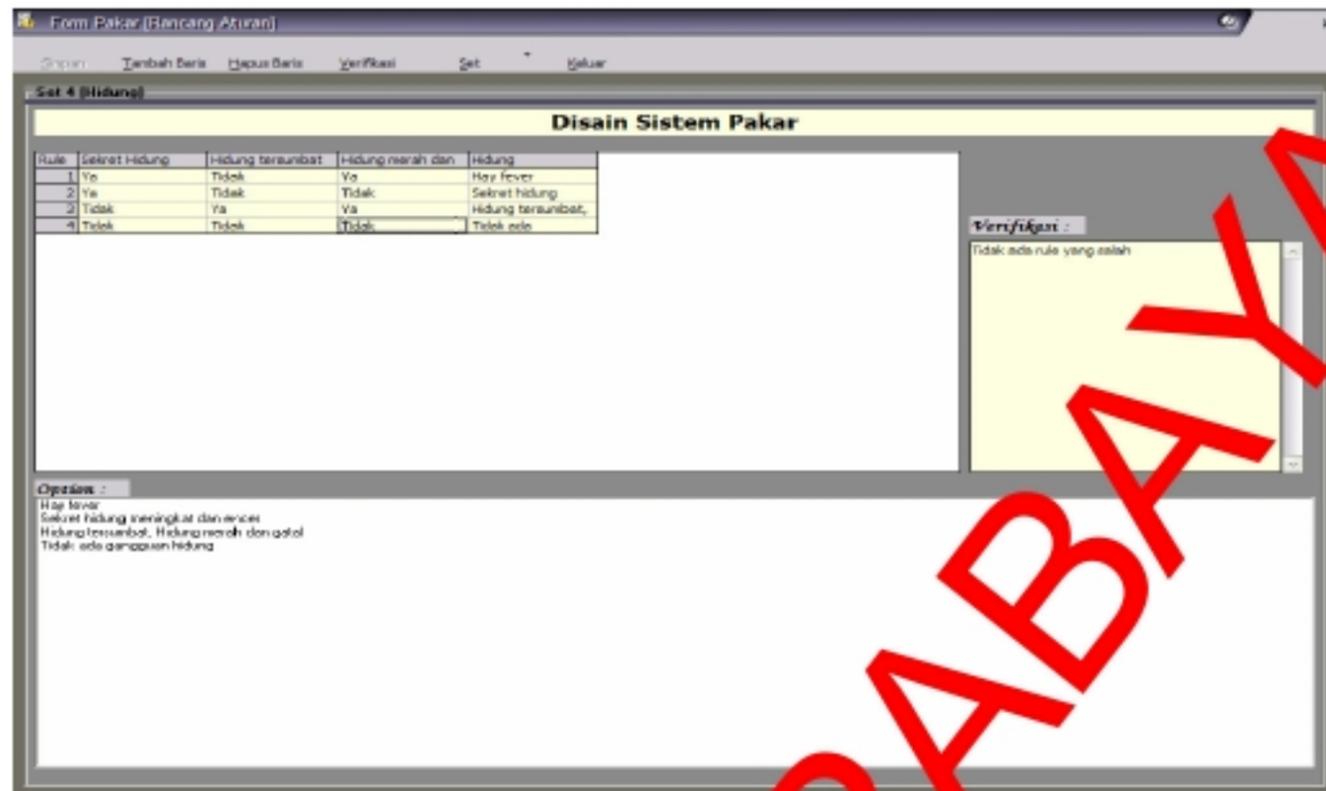
data-data tersebut diambil dari beberapa macam permasalahan yang berbeda. Hal ini dilakukan agar dapat diketahui bahwa proses Verifikasi mampu menghasilkan *Rule-Rule* yang sesuai dengan desain permasalahan. Selain itu testing pada sistem ini dilakukan untuk memastikan apakah sistem telah berjalan dengan baik atau tidak

4.2.1 Testing Kinerja Sistem

Testing ini dilakukan untuk menguji apakah kinerja sistem dalam melakukan Verifikasi dan proses inferensi. Tujuan pengujian ini untuk mengetahui apakah *Rule-Rule* yang dihasilkan dari proses Verifikasi tidak terjadi *Redundant Rules*, *Conflicting Rules*, *Circular Rules*, *Subsumed Rules*, *Unnecessary IF Conditions* dan *Dead end Rules*. Dan tujuan dari pengujian proses inferensi adalah apakah sistem dapat membantu *User* dalam menentukan penyakit flu burung beserta saran pengobatan. Dari serangkaian uji coba yang telah dilakukan maka didapatkan hasil sebagai berikut yaitu:

Kasus 1:

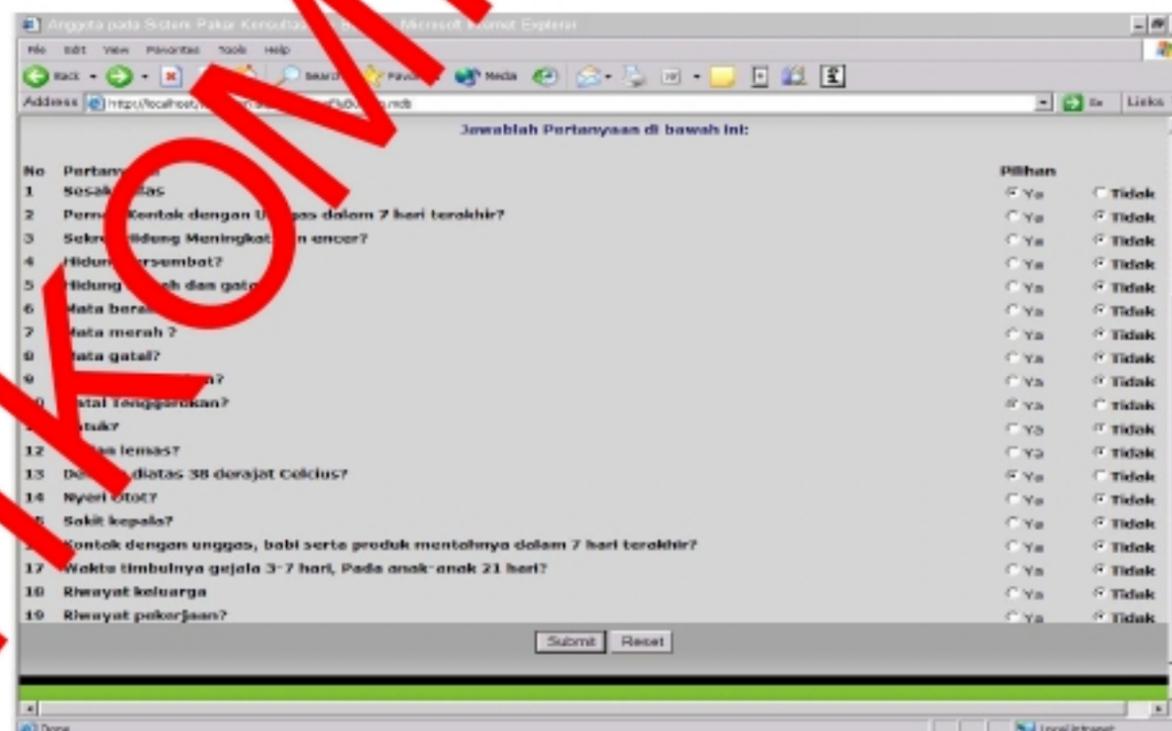
Pada set 4 terdapat parameter sekret hidung meningkat dengan *Option* ya dan tidak, parameter hidung tersumbat dengan *Option* ya dan tidak, parameter hidung merah dan gatal dengan *Option* ya dan tidak, parameter hidung dengan *Option* hay fever, sekret hidung meningkat, hidung tersumbat dan gatal, kemudian tidak ada gangguan berdasarkan hasil *Reduced Decision Table set 3* hasil verifikasi *Rulanya* adalah seperti pada Gambar 4.26.



Gambar 4.26 Hasil Verifikasi Rule Set 4

Kasus 2:

Pasien dengan gejala gatal tenggorokan, sesak napas, demam diatas 30 derajat celcius, hidung tersumbat merupakan gejala penyakit flu burung Suspek dapat dilihat pada Gambar 4.27 berikut:



Gambar 4.27 Gejala Penyakit Flu Burung Suspek

Dan hasil inferensinya dapat dilihat pada Gambar 4.28 berikut ini:



Gambar 4.28 Hasil proses inferensi